Return To Agenda Page

Useable Multicore Implementations in Embedded Applications

October 26, 2005

Toby Foster
System Architect
Freescale Semiconductor
toby.foster@freescale.com

freescale™ semiconductor

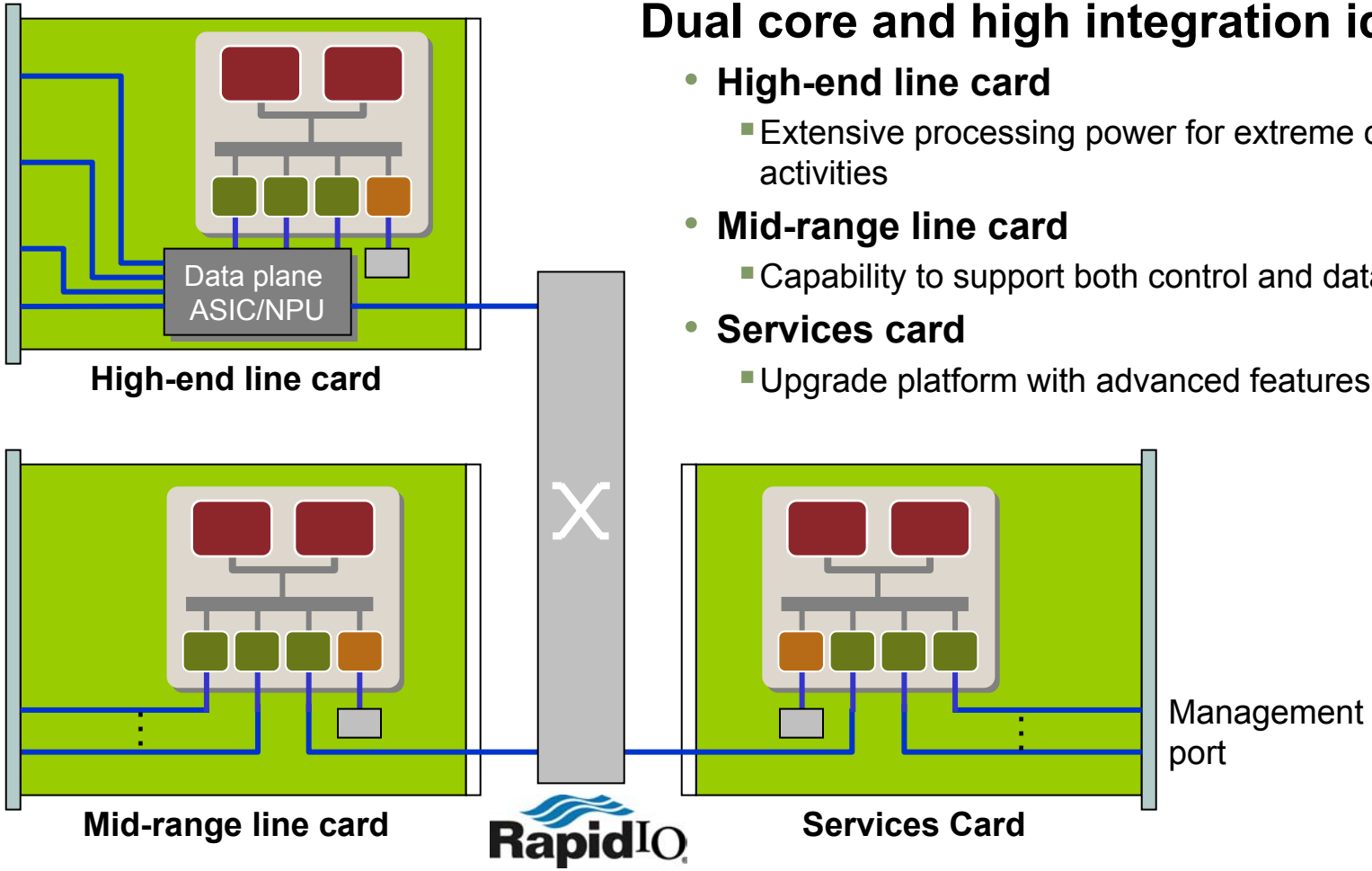
Agenda

- **Dual core application flexibility**
- **Implementing a usable dual core device**
  - Block diagram
  - SMP considerations
  - Asymmetric MP considerations

Agenda

- **Dual core application flexibility**
- **Implementing a usable dual core device**
  - Block diagram
  - SMP considerations
  - Asymmetric MP considerations

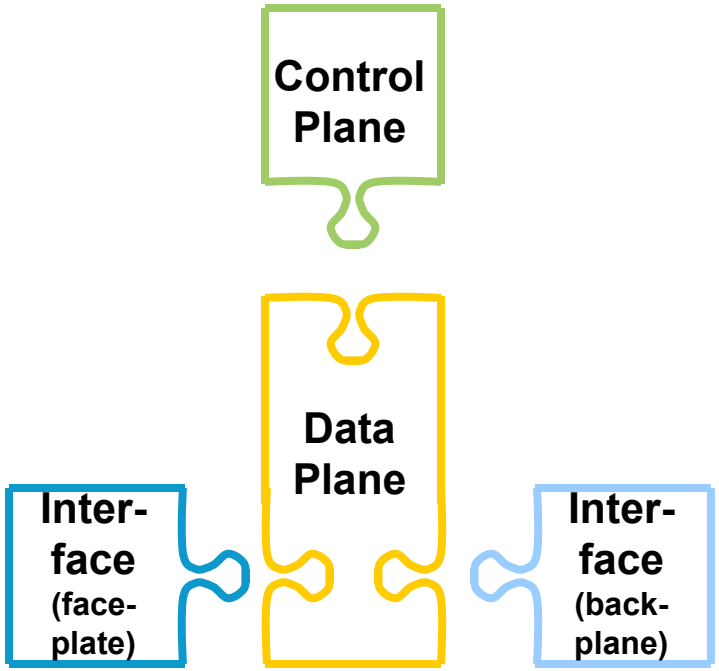
Dual Core Example Applications



Dual core and high integration ideal for

- **High-end line card**
  - Extensive processing power for extreme control plane activities
- **Mid-range line card**
  - Capability to support both control and data
- **Services card**
  - Upgrade platform with advanced features

Generic Architecture



Network protocol processing
Policy
Network Management
Signaling
Topology Management

Traffic management
Data transformation
Classification
Parsing

Ethernet
SONET
T1/E1

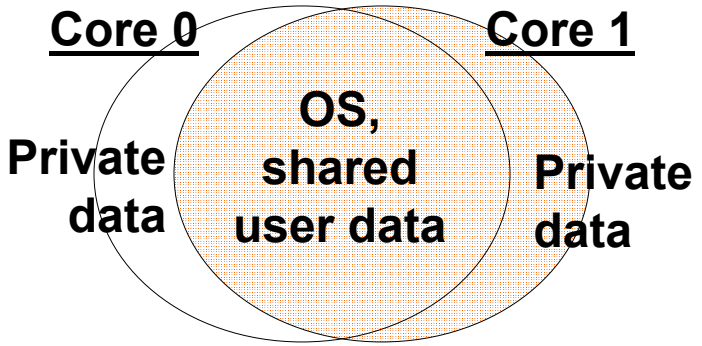
PCI
PCI Express
RadipIO

Multiprocessing Configurations

Symmetric Multiprocessing

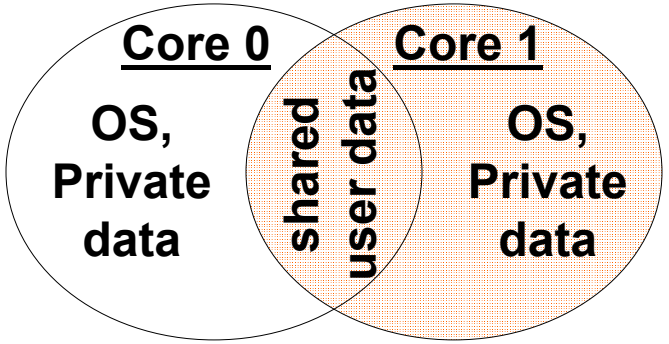
- Homogenous OS support
- High-performance option
- Software transparency
- Cores share address space for OS and data
- Resource sharing handled by OS
- Dynamic load balancing by OS

Memory Map Overlap

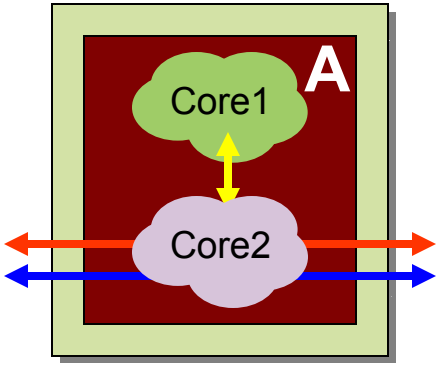


Asymmetric Multiprocessing

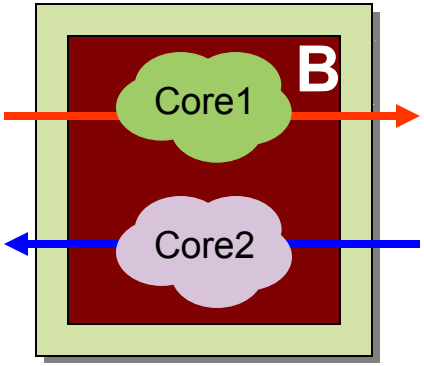
- Heterogeneous OS support
- Two separate OS or two copies of one non-SMP OS
- Collapse two processors into one
- Task offload or division of labor
- Operating systems, data reside in different address spaces
- Resource sharing handled by user
- Static load balancing



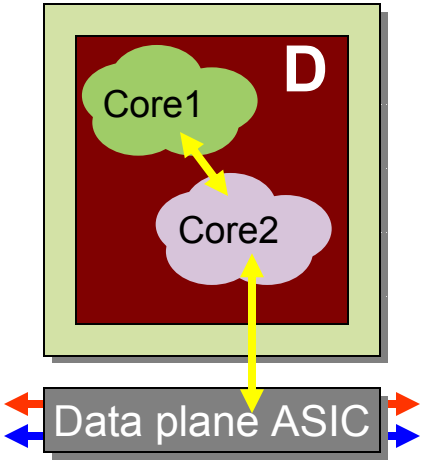
Usage of a Dual Core Device



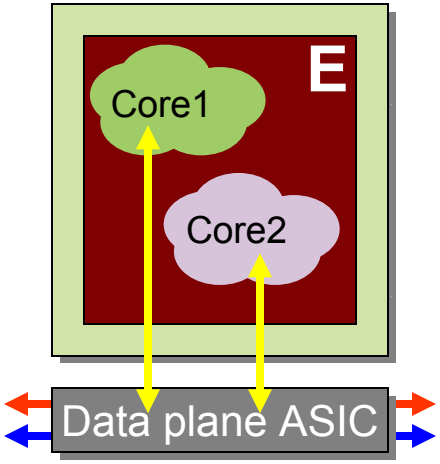
One core handles data plane, one control plane



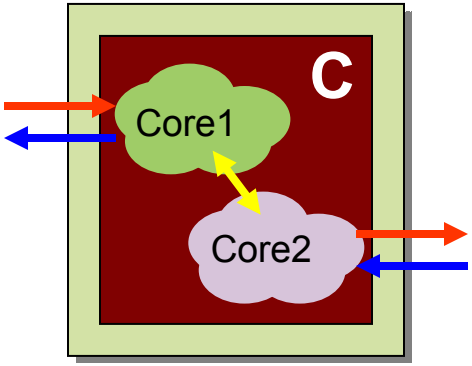
Each core handles one direction of data plane



Task offload

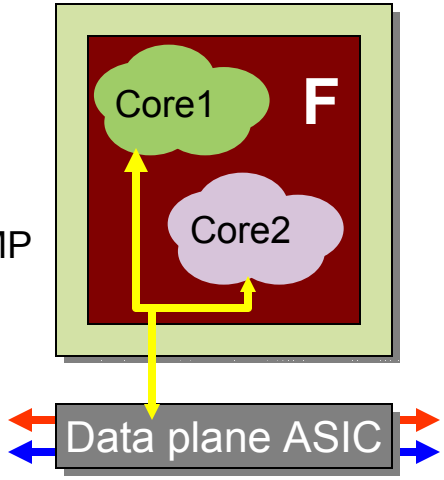


Each core handles a separate aspect of control plane



Network and disk partioning

SMP



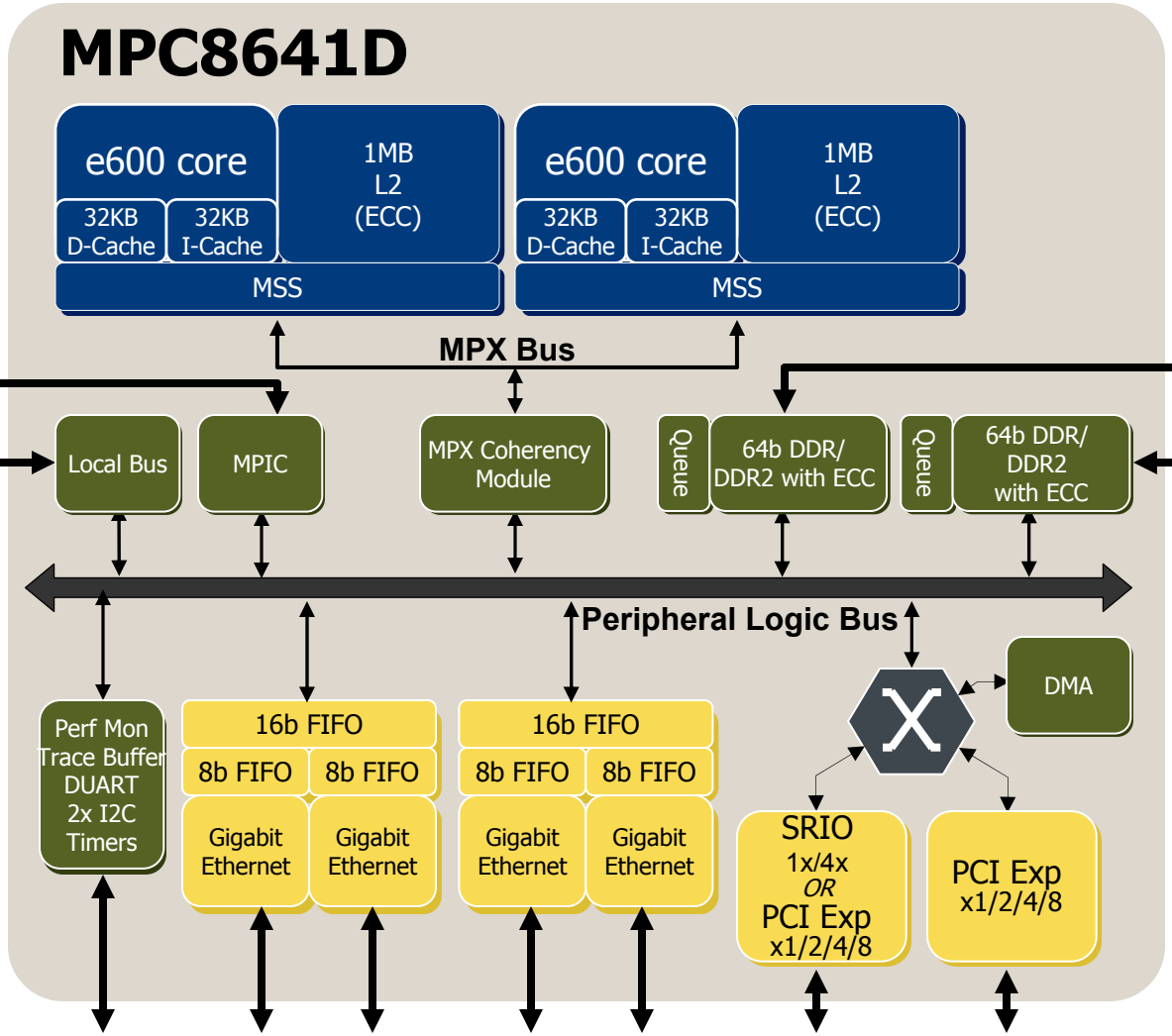
Mid Range

High End

Agenda

- **Dual core application flexibility**
- **Implementing a usable dual core device**
  - Block diagram
  - SMP considerations
  - Asymmetric MP considerations

MPC8641D – A Usable Dual Core Solution

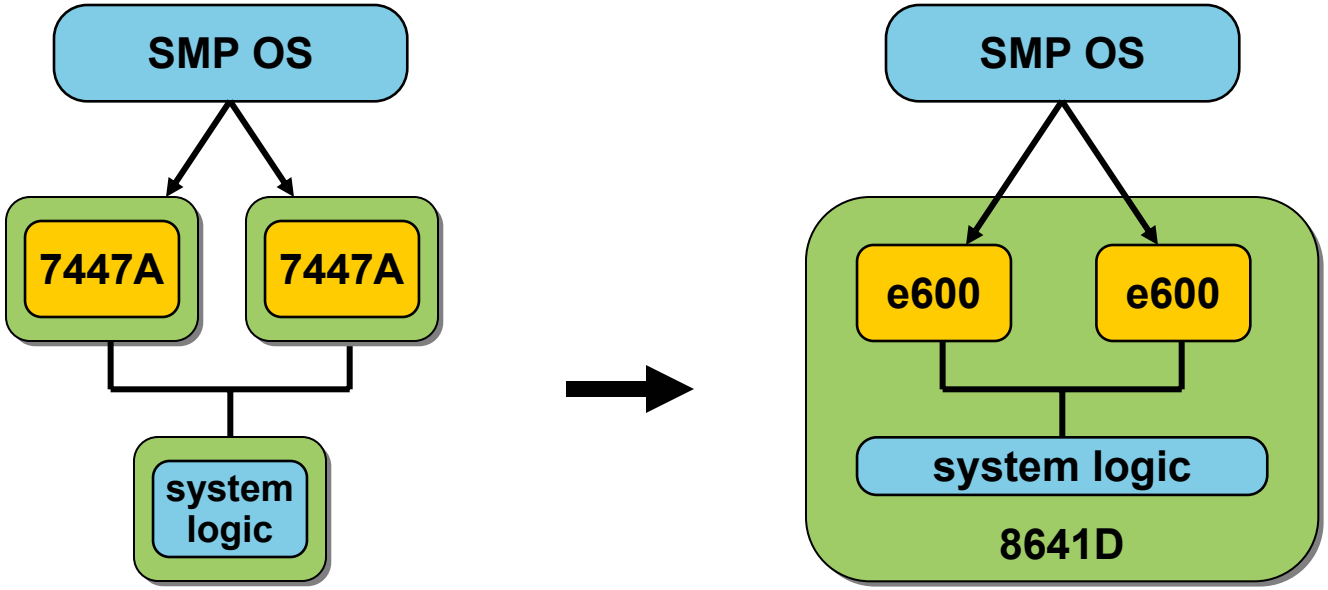


- **Dual e600 PowerPC cores (1.0-1.67GHz)**
  - Altivec
  - 1MB L2 Cache w/ECC per core
  - 36bit physical addressing
- **System Unit**
  - 64b DDR/DDR2/FCRAM2 w/ECC
  - 4x 10/100/1000 Ethernet Controllers w/TCP Offload, QoS, and 2x 16b or 4x 8b FIFO-mode (up to 12.8 Gbps)
  - DUART, 2x I^2C, Multiprocessor Interrupt Controller with Timers, Trace Buffer
- **High-speed Interfaces**
  - 1x/4x SRIO (2.5GB/s) and x1/x2/x4/x8 PCI-Express (4GB/s)
  - OR two x1/x2/x4/x8 PCI-Express (8GB/s)
- **90nm Technology – 105C Tj**

Agenda

- **Dual core application flexibility**
- **Implementing a usable dual core device**
  - Block diagram
  - SMP considerations
  - Asymmetric MP considerations

Symmetric Multiprocessing Considerations

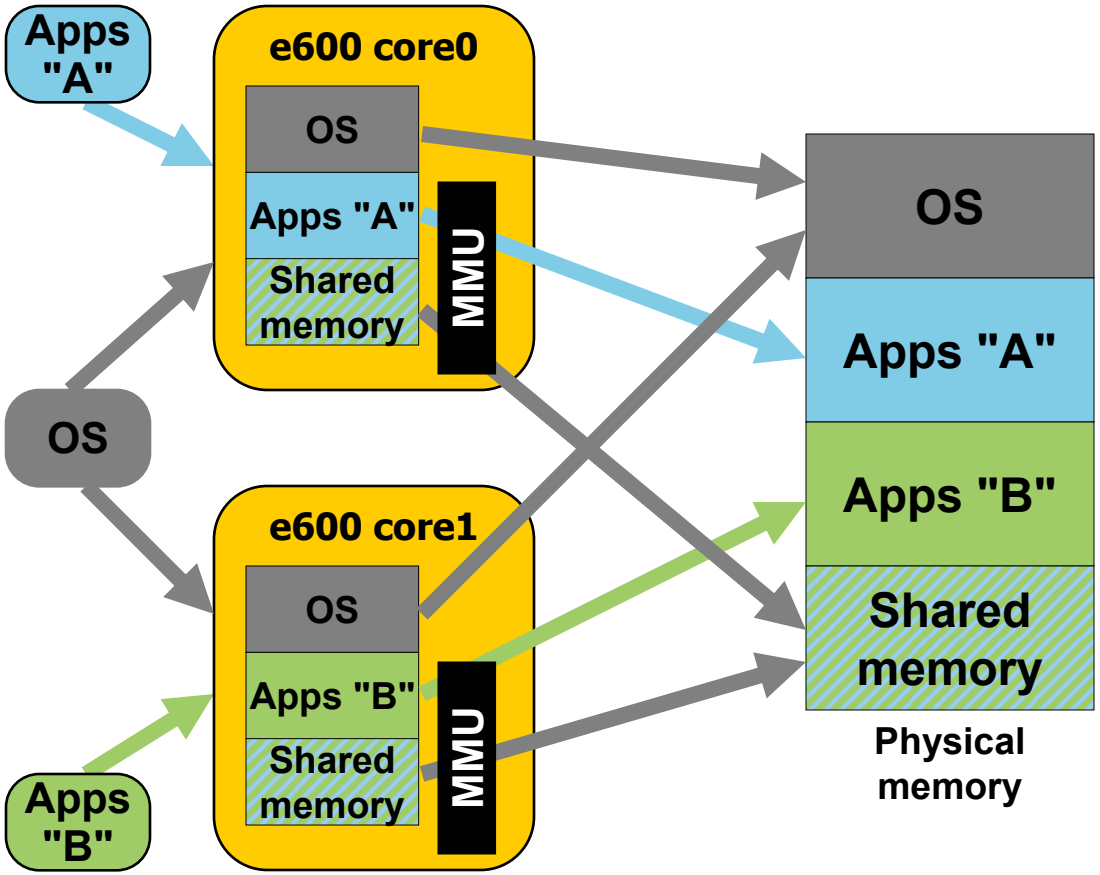


- One OS kernel image in physical memory
- Both cores execute the same OS kernel image
- SMP OS owns all of the resources
- Linux, QNX, *BSD only embedded SMP OSes

SMP Memory Organization

The OS kernel resides at physical memory address 0, addressable by both cores

The MMU relocates applications and shared memory appropriately



What is Coherency?

Consistent view of memory across multiple agents

- Buffer descriptors and data buffers updated by processor(s) as well as external agent(s)

Software-managed coherency

- Processor overhead to keep track of “who owns what when”

Hardware-managed coherency

- Each processor’s hardware ensures consistency of shared data by snooping other agents’s broadcasts on the system bus

Performance Features of HW-Coherency

Coherency protocol

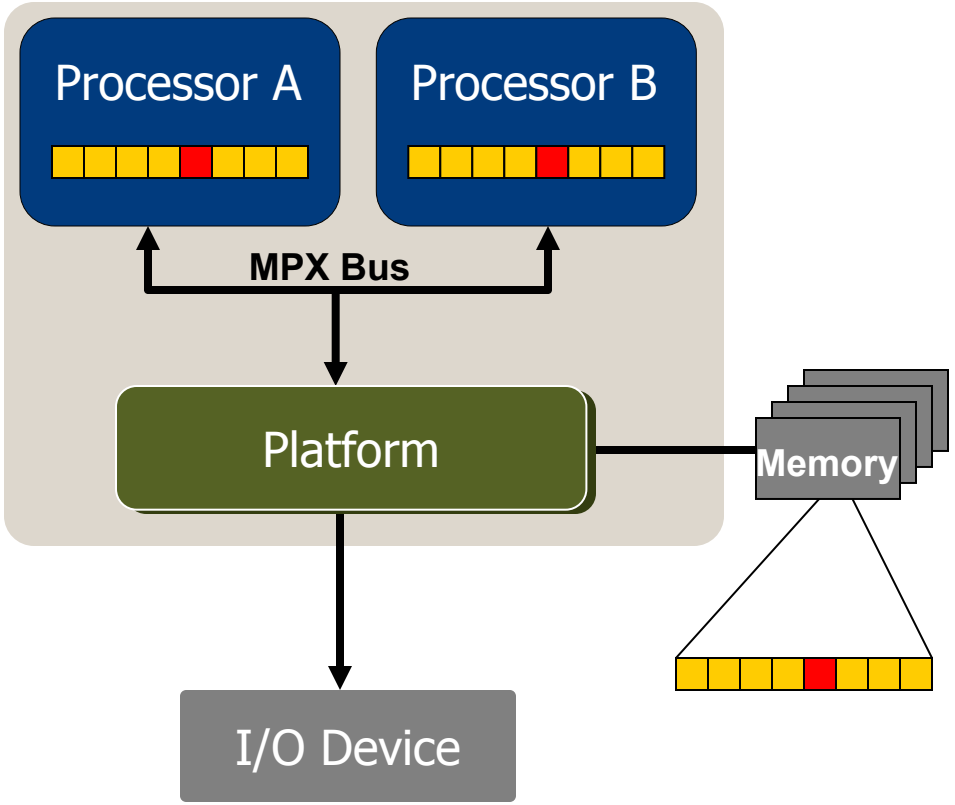
- MEI
- MESI

Update mechanism

- Push
- Intervention

Cache Tags

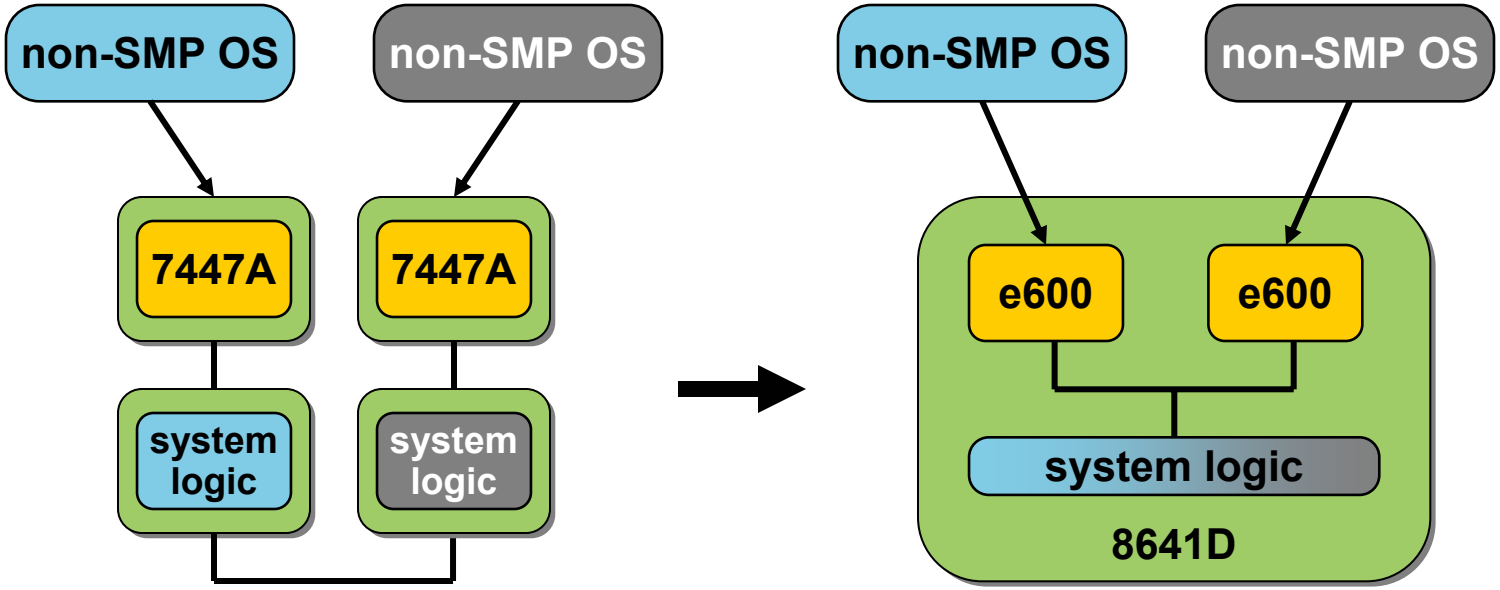
- Single-ported
- Dual-ported



Agenda

- **Dual core application flexibility**
- **Implementing a usable dual core device**
  - Block diagram
  - SMP considerations
  - Asymmetric MP considerations

Asymmetric Integration



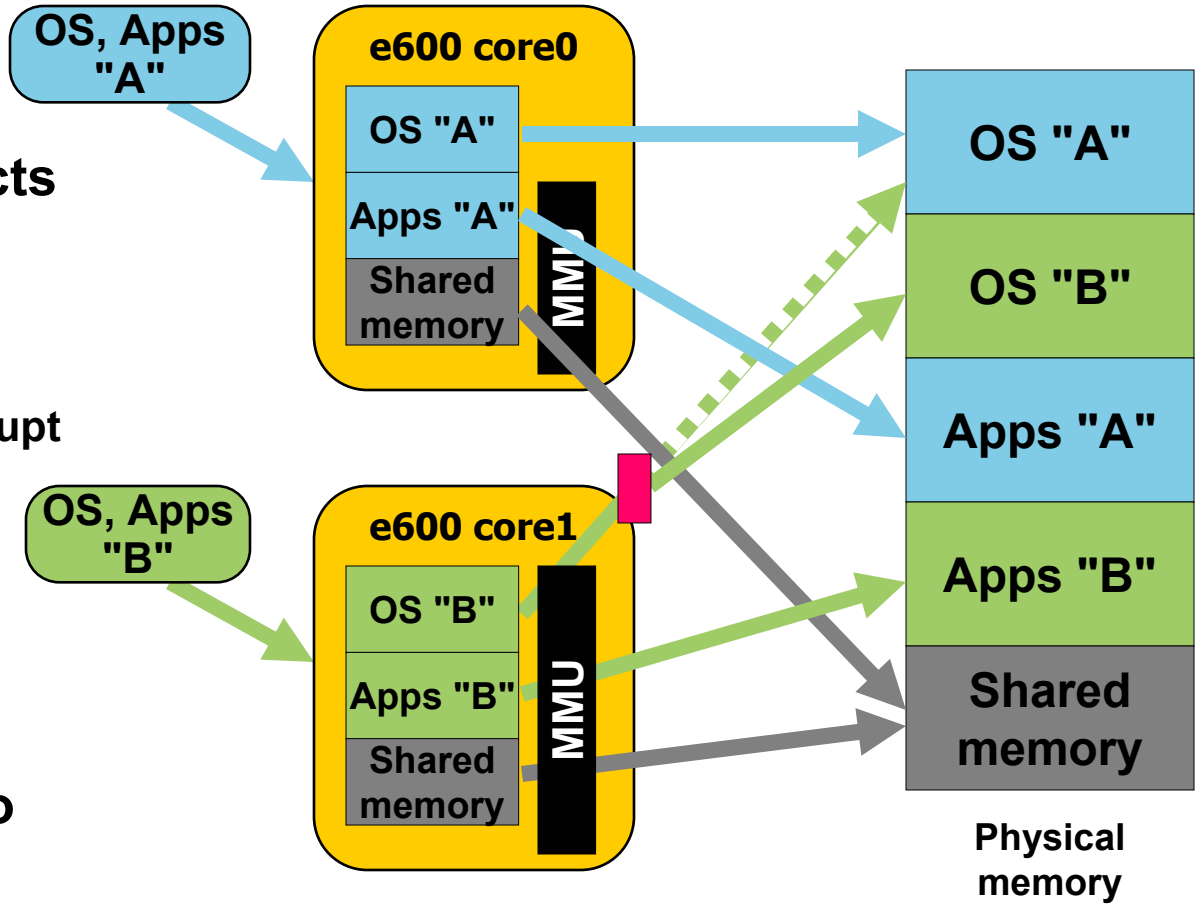
- Two OS kernel images in physical memory
- Each core executes a separate OS kernel image
- Non-SMP OSes must cooperate in sharing resources
- VxWorks, OSE, Integrity, Jaluna-1, many others

Asymmetric MP Memory Organization

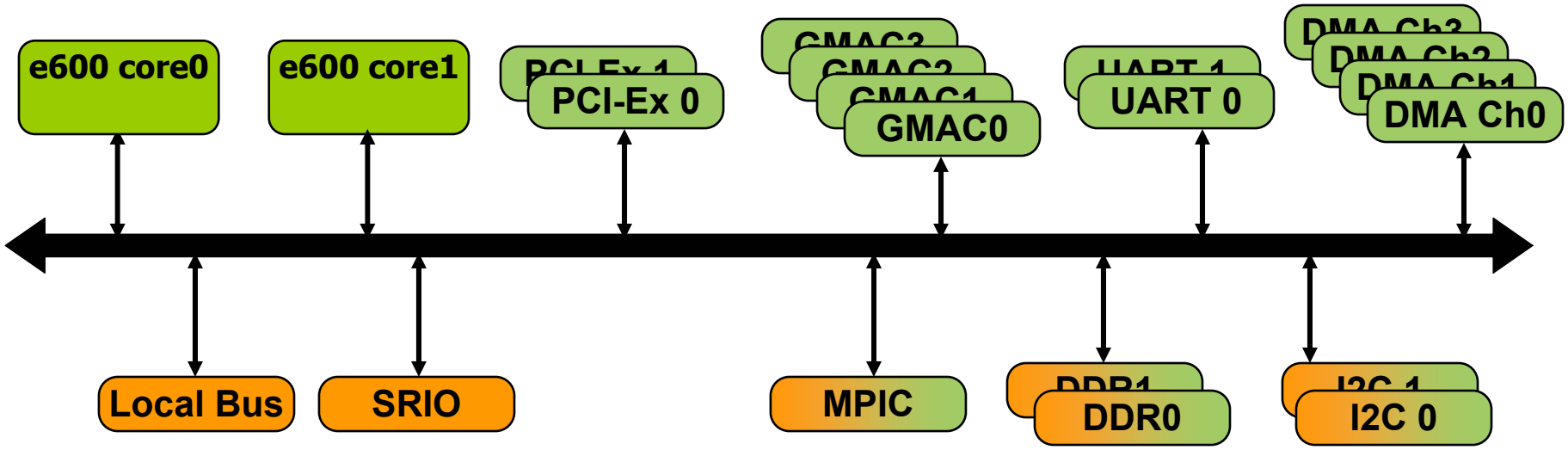
Each OS kernel expects to control physical memory beginning at address 0

- Each wants its own interrupt vectors
- The MMU can relocate applications and shared memory appropriately

The 8641D includes a hardware translator to relocate physical address 0 for core1



Resources: Shared or Multiple Instances

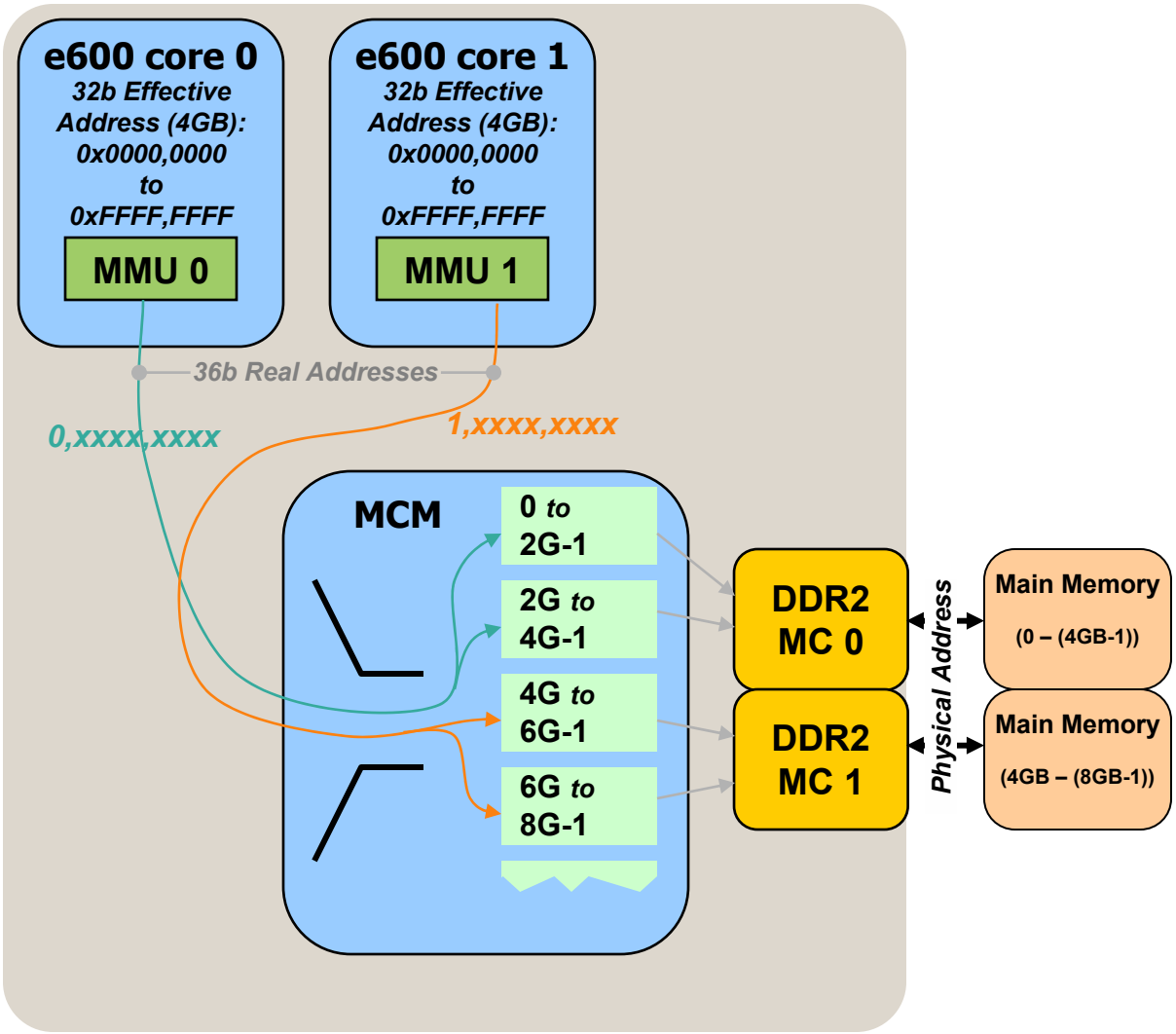


Multiple resource instances

Shared Resource

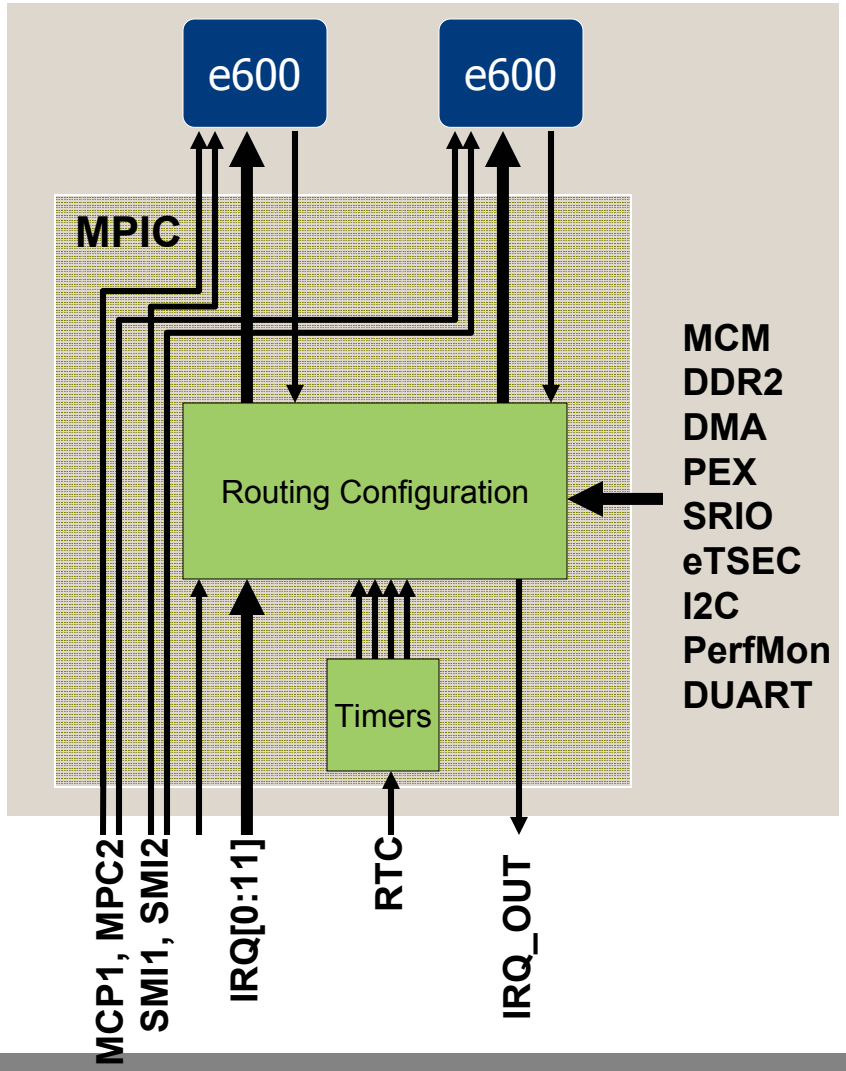
Partially shared or multiple instances in some circumstances

Private Memory Controllers



- **A core can be assigned to a MC**
- **Within core**
  - Map address to MCM window
- **Within MCM**
  - Map window to memory controller
- **Possible to assign peripherals to one core or the other in similar way**
  - ATMUs create addresses that fall into window of target MC

Multi-Processor Interrupt Controller



Sources
- 17 external
- 32 internal

Multi-core capability
- Per core interrupt routing
- Dedicated core interrupts
- Intercore communication
- Register duplication

Prioritization
- 16 levels
- Fully nested

Modes
- Pass through and Mixed

Four global timers
- MPX Clock
- RTC

OpenPIC compliant

Memory Protection

- Cores cannot access each other's internal registers or caches
- Programming models for on-chip I/O are independent
- I/O devices in separate 4kB pages
- Main memory can be allocated to one core or another
- Per-Page/BAT configurations
  - **No access, read only, read/write, no-execute, user/supervisor mode access**
- MCM Local Address Windows define valid memory ranges
- ATMUs control what addresses enter/exit device

Other features supporting dual OS usage of 8641D

- Ability to start and stop clocks on both cores separately or simultaneously via JTag
- Ability to reset each core separately via JTag
- Ability to release the two cores from reset separately or simultaneously
- Low-skid, correlated breakpoint capability between cores
- Inter-core soft reset capability via interrupt controller

Summary

A well thought out dual core device…

- **Efficient SMP implementation**
- **Features needed to enable Asymmetric MP**

… leads to a flexible device suitable for many embedded applications